Claims

What is claimed is:

1. A method for providing multimedia functionality in a homogeneous multiprocessor environment comprising the steps of:

Sub A1

queuing tasks;
identifying available processing resources in the homogeneous multiprocessor environment;
allocating the available processing resources among the tasks;
providing to the available processing resources functional programs and initial data corresponding to the tasks;
performing the tasks using the available processing resources to produce resulting data.

2. The method of claim 1 wherein a plurality of processors of the homogeneous multiprocessor environment are capable of executing a first instruction of a first instruction set and a second instruction of a second instruction set.

3. The method of claim 2 wherein the first instruction and the second instruction share an identical bit pattern but perform different operations.

4. The method of claim 3 wherein a first processor of the plurality of processors executes an input/output kernel program, the input/output kernel program including a first portion expressed using the first instruction set and a second portion expressed using the second instruction set.

5. The method of claim 3 further comprising the step of:

converting a functional program of the functional programs expressed using the first instruction set to an equivalent functional program expressed using the second instruction set.

Sub A1

6. The method of claim 3 further wherein the tasks comprise:
x86 processing;
graphic image processing;
video processing;
audio processing; and
communication processing.

7. The method of claim 3 further comprising the step of:
receiving the initial data from a first input/output device.

8. The method of claim 3 further comprising the steps of:
passing the resulting data to a first input/output device.

9. The method of claim 8 wherein the step of passing the resulting data to the first input/output device further comprises the step of:

passing the resulting data through an intermediary device, wherein the intermediary device is coupled to the first input/output device and to a second input/output device.

10. The method of claim 9 wherein the step of passing the resulting data through an intermediary device, wherein the intermediary device is coupled to the first input/output device and to a second input/output device further comprises the step of:

automatically adapting to a reallocation of the available processing resources among the tasks.

11. The method of claim 8 wherein the step of passing the resulting data to a first input/output device further comprises the step of:

passing the resulting data to a mixed-signal device.

12. The method of claim 3 wherein the step of allocating the available processing resources among the tasks is dynamically adjusted.

Sub A1

13. Apparatus comprising:

a plurality of processors coupled to a bus;

an input/output interface coupled to the bus;

a plurality of input/output devices coupled to the input/output interface, the plurality of processors processing program code configured to perform a plurality of tasks, the program code comprising:

program code configured to cause a first portion of the plurality of processors to interact with a first input/output device of the plurality of input/output devices;

program code configured to cause a second portion of the plurality of processors to interact with a second input/output device of the plurality of input/output devices;

program code configured to cause a second portion of the plurality of processors to emulate a specific microprocessor instruction set.

14. The apparatus of claim 8 further comprising:

kernel program code configured to dynamically allocate the processing of the program code among the plurality of processors.